

# SMW for Programming/Teaching Context-Aware AAL Agents

Nicole Merkle (FZI)

29.09.2016

13th SMWCon Frankfurt am Main, Germany



# Retrospective-What I talked about the last time?

## At SMWCon Fall 2015, Barcelona, Spain:

- Ambient Assisted Living (AAL)
- Web of Things (WoT) of approach.
- Modelling of AAL domain by SMW.
- Sherlock Engine as assistive agent.



## Today:

- SMW as programming tool for assistive, context-aware agents, e.g Sherlock.



# Why is this relevant?

# Lost in the space of many APIs, Protocols and Programming Languages



# Retrospect: The Web of Things

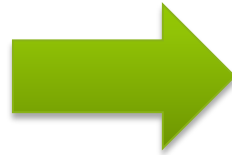
## Things of Interest



IoT Devices

## Thing Descriptions

**URI:** <http://ont.Actor1>  
**Action:** <http://ont.SwitchOff>  
**Action:** <http://ont.SwitchOn>  
**Event:** <http://ont.LampOn>  
**Event:** <http://ont.LampOff>  
**Property:** <http://ont.SwitchActor>  
**Property:** „white“



Machine-understandable

Agents



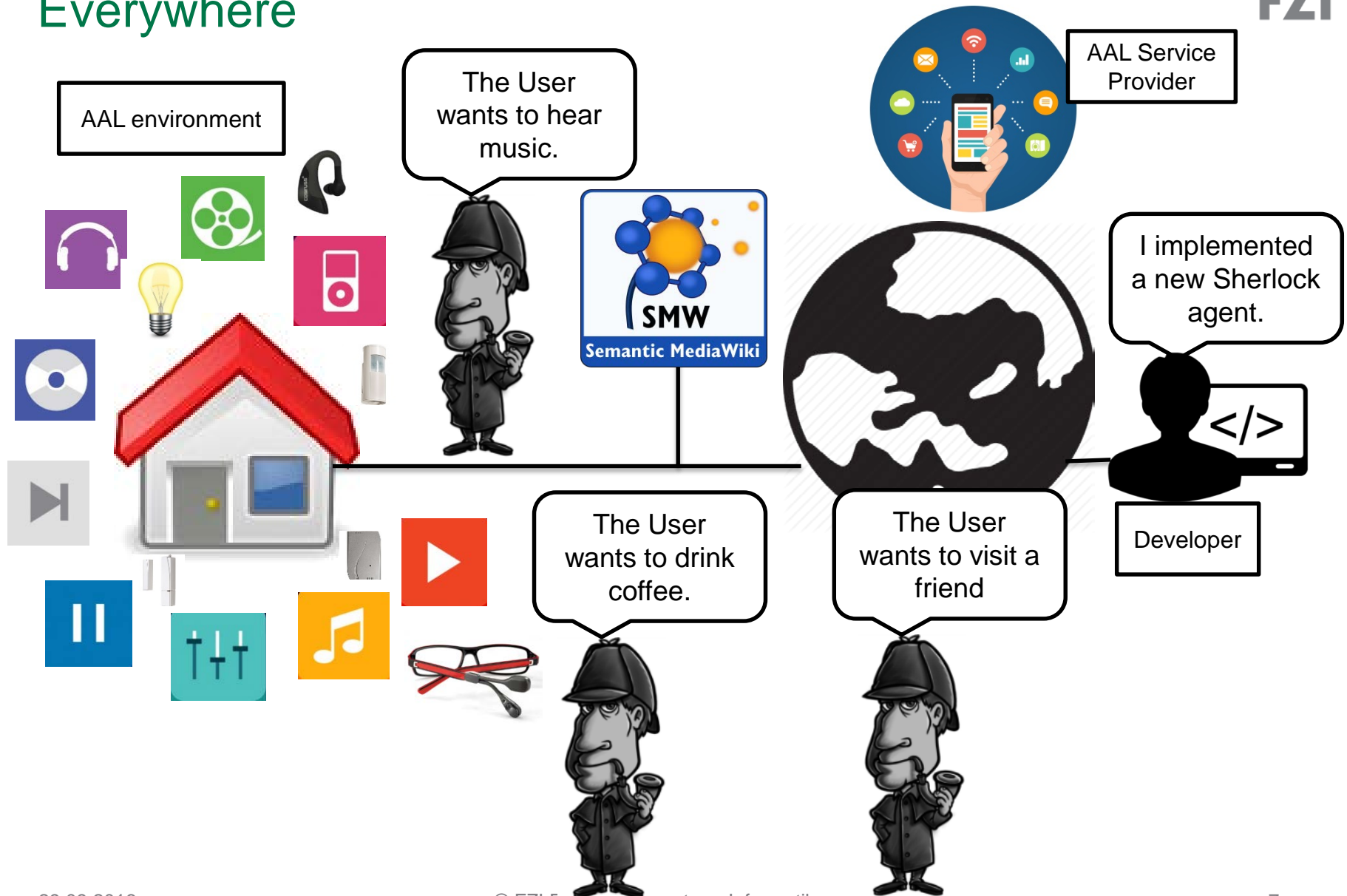
Web-Protocols

# Rule-Based Sherlock-Agent as User Assistant

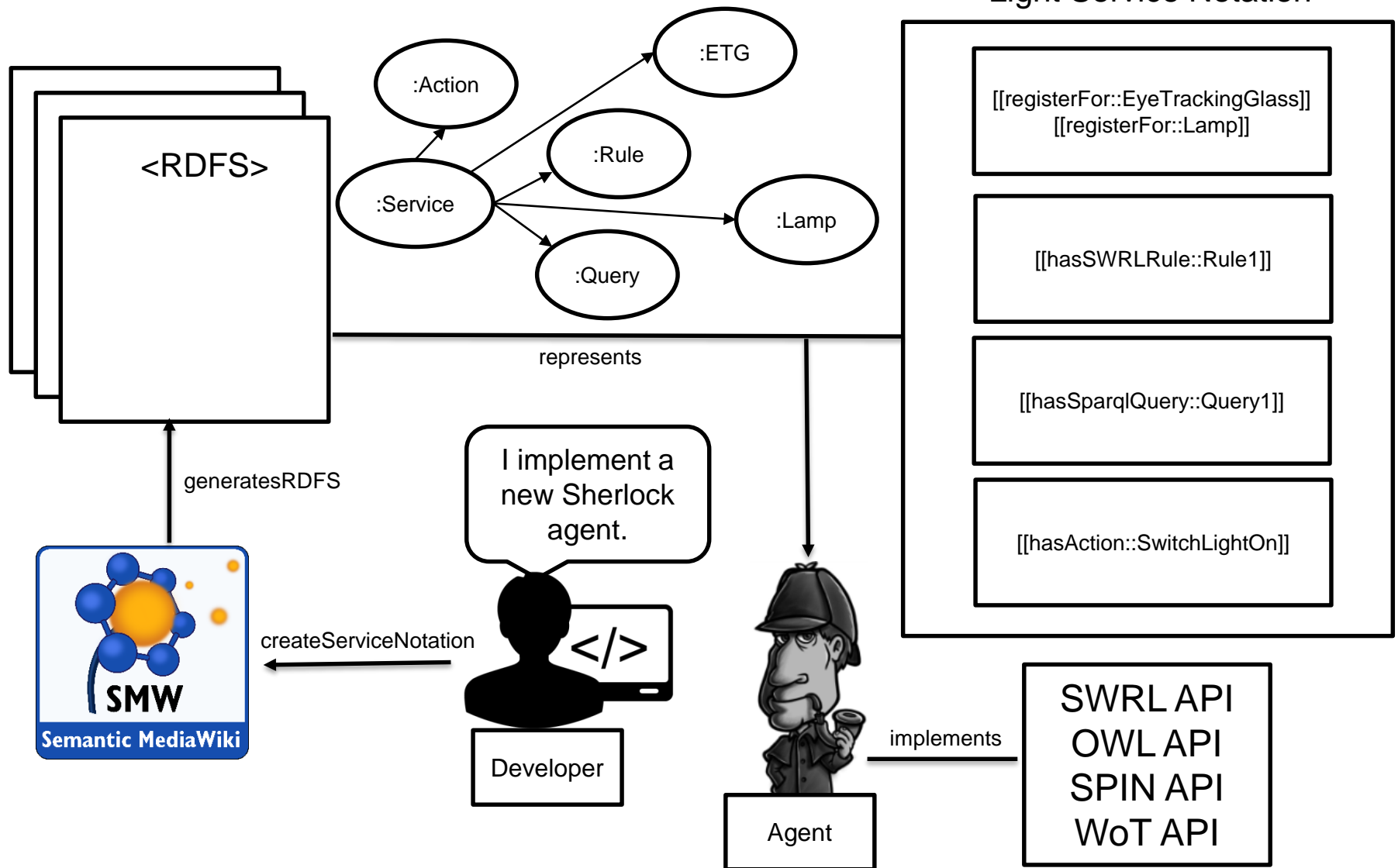




# A Vision: SMW Programmed Sherlock Agents Everywhere



# Sherlock's Knowledge Representation





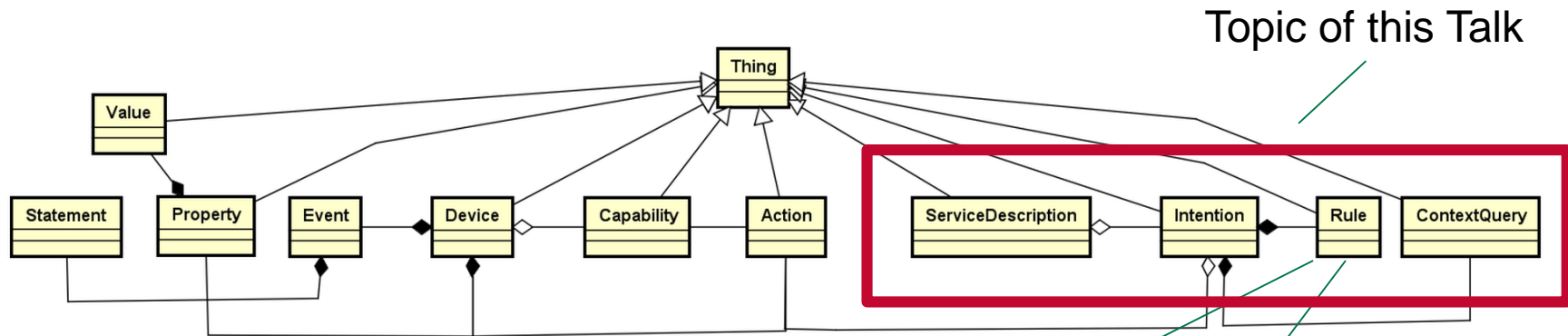
# Rule-based languages

- **Semantic Web Rule Language** (abbr. **SWRL**):
  - First order logic. Conjunctions of Terms.
  - No Negation.
  - Structure: Premise -> conclusion
  - Terms as Tuples: predicate(?subject, ?object) or Class(?instance)
  - Expressing Triples: subject, predicate, object
- **SPARQL Protocol And RDF Query Language**: Querying RDF(S) graphs
  - SELECT, CONSTRUCT, ASK queries
  - Filter Rules by triples
- **SPARQL Inferencing Notation** (abbr. **SPIN**): Save SPARQL queries in RDF(S) representation



Idea: Embed SWRL rules  
SPARQL queries and SPIN in  
SMW pages

# AAL Domain Classes (Things of Interest)



## SWRL Rule



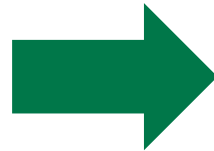
```

AssistedPerson(?u) ∧ Lamp(?l) ∧ hasState(?l, "off") ∧ AmbientLight(?a)
∧ hasState(?a, ?s) ∧ swrlb : lowerThan(?s, 600) ∧ wears(?u, ?e)
∧ EyeTrackingGlass(?e) ∧ hasInFocus(?e, ?l) ∧ Intention(SwitchLightOnIntention)
∧ hasAction(SwitchLightOnIntention, SwitchOnAction)
∧ hasAction(?l, SwitchOnAction) → hasIntention(?u, SwitchLightOnIntention)
    
```

A device of type lamp has the state off and an Ambient light sensor has a state lower than 600 lux. The Assisted Person wears an Eye Tracking Glass and this EyeTracking Glass has a thing of type lamp in focus. A lamp can be turned on and off.  
Conclusion the assisted person wants to turn the lamp on.

# SPARQL Query Example

Give me all temperature sensors which are in the Living Room and which state is greater than 25 celsius degree.



```
PREFIX wiki: <http://localhst/mywiki#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?sensor
WHERE {
  ?sensor wiki:hasCelsiusDegree ?degree.
  ?sensor rdf:type wiki:Category:TemperatureSensor.
  ?sensor wiki:isInRoom wiki:Living Room.
  FILTER(?degree < 25)
}
```



Context: [Bearbeiten]  
<https://koralle27.fz.fraunhofer.de/aiacsys/ontology#intention>

Label: [Bearbeiten]  
 Lampe einschalten.

Action: [Bearbeiten]  
[Switch Light on](#)

Vorhergehende Intentionen [Bearbeiten]  
[Show\\_Information\\_On\\_Display](#), [Show\\_Information\\_On\\_Smartphone](#), [Show\\_Information\\_On\\_Glass](#)

Regeln: [Bearbeiten]  
 Action(Switch\_Light\_On) \* AmbientLightSensor(?ambie) \* AssistedPerson(?u) \* EyeTrackingGlass(?e) \* Intention(Switch\_Light\_On\_intention) \* Lamp(?l) \* HasFunctionality(?f, Switch\_Light\_On) \* HasAction(Switch\_Light\_On\_intention, Switch\_Light\_On) \* HasInFocus(?f, ?l) \* HasState(?ambie, ?ambientState) \* HasState(?l, "OFF") \* swmb greaterThan(?ambientState, 500) -> HasIntention(?u, Switch\_Light\_On\_intention)

Context Query [Bearbeiten]  
 PREFIX rdf: <<http://www.w3.org/1999/02/22-rdf-syntax-ns#>>  
 PREFIX rdfs: <<http://www.w3.org/2000/01/rdf-schema#>>  
 PREFIX owl: <<http://www.w3.org/2002/07/owl#>>  
 PREFIX wiki: <[http://koralle32.fz.fraunhofer.de/aiacsys/index.php/Spezial\\_URI-Auff%C3%B6ser/](http://koralle32.fz.fraunhofer.de/aiacsys/index.php/Spezial_URI-Auff%C3%B6ser/)>  
 PREFIX wikurt: <<http://koralle32.fz.fraunhofer.de/aiacsys/index.php/>>  
 PREFIX foaf: <<http://xmlns.com/foaf/0.1/>>  
 PREFIX xsd: <<http://www.w3.org/2001/XMLSchema#>>  
 PREFIX swit: <<http://semantic-mediawiki.org/swit/1.0/>>  
 select ?inventory ?sensor  
 where {  
 ?inventory wiki:Attribut-3AIsInRoom ?room.  
 ?inventory rdf:type wiki:Kategorie-3ALamp.  
 ?room wiki:Attribut-3AHasInventory ?inventory.  
 ?room wiki:Attribut-3AHasSensor ?sensor.  
 ?sensor rdf:type wiki:Kategorie-3AAmbientLightSensor  
 }

Kategorie: Intention

Embedded queries and rules

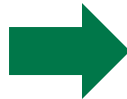
# SPIN Example

```

{{SPIN
|description=
|querytype=
|query=
}}
</pre>
Klick auf „Bearbeiten“, um den Quelltext der Vorlage anzusehen.
</noinclude><includeonly>
== Description of the SPIN Rule ==
{{{description}}}

== Generated SPIN Rule ==
{{#set:
HasSpinRule=[ a
sp:{{{querytype}}}} ;
sp:text ""
{{{query}}}
""
]
}}

```



## Bearbeite SPIN: Test Rule

**Description:**

**Query Type:**

**Query:**



## Test Rule

### Description of the SPIN Rule [\[Bearbeiten\]](#)

Show just people with an age over 18 years.

### Generated SPIN Rule [\[Bearbeiten\]](#)

```

[ a
sp:Ask ;
sp:text ""

    1. Show just people with an age over 18 years.

ASK WHERE {
?this my:age ?age .
FILTER (?age < 18) .
}
""
]

```

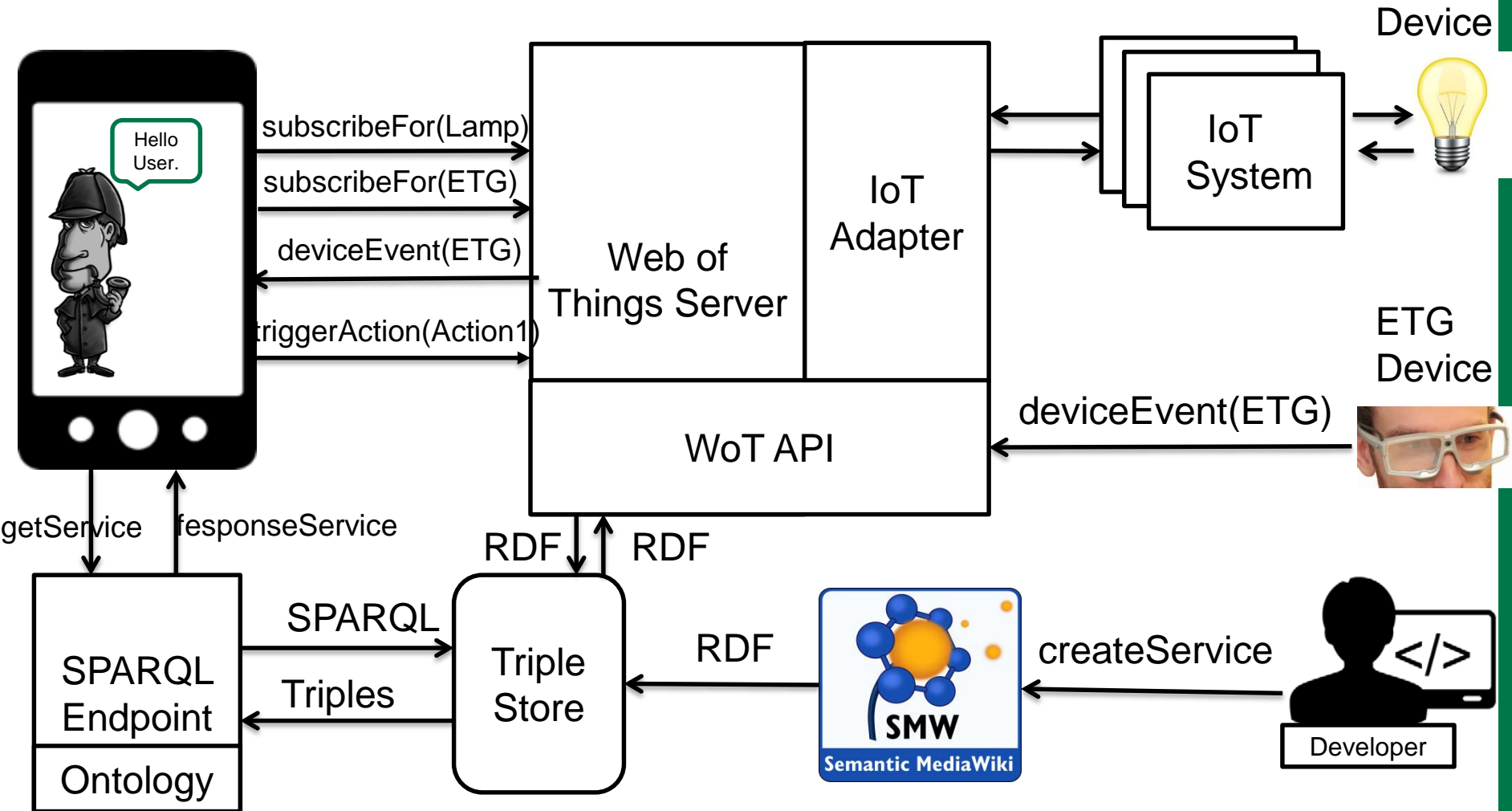


```

[ a
sp:Ask ;
sp:text ""
# Show just people with an age over 18
years.
ASK WHERE {
?this my:age ?age .
FILTER (?age < 18) .
}
""
]

```

# The Context-sensitive Sherlock Agent



# Sherlock shall be able to learn

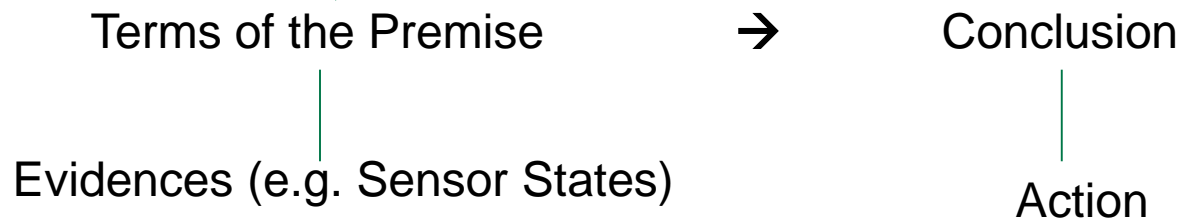
SMW provides **static information/knowledge** about the user, but the **context** of the user is **dynamic** and **uncertain**.

**Conclusion:** The **preferences and intentions** must be **learned** and the **rules** must be **adjusted**. Furthermore, in **uncertain situations** the **user** has to be **asked**.



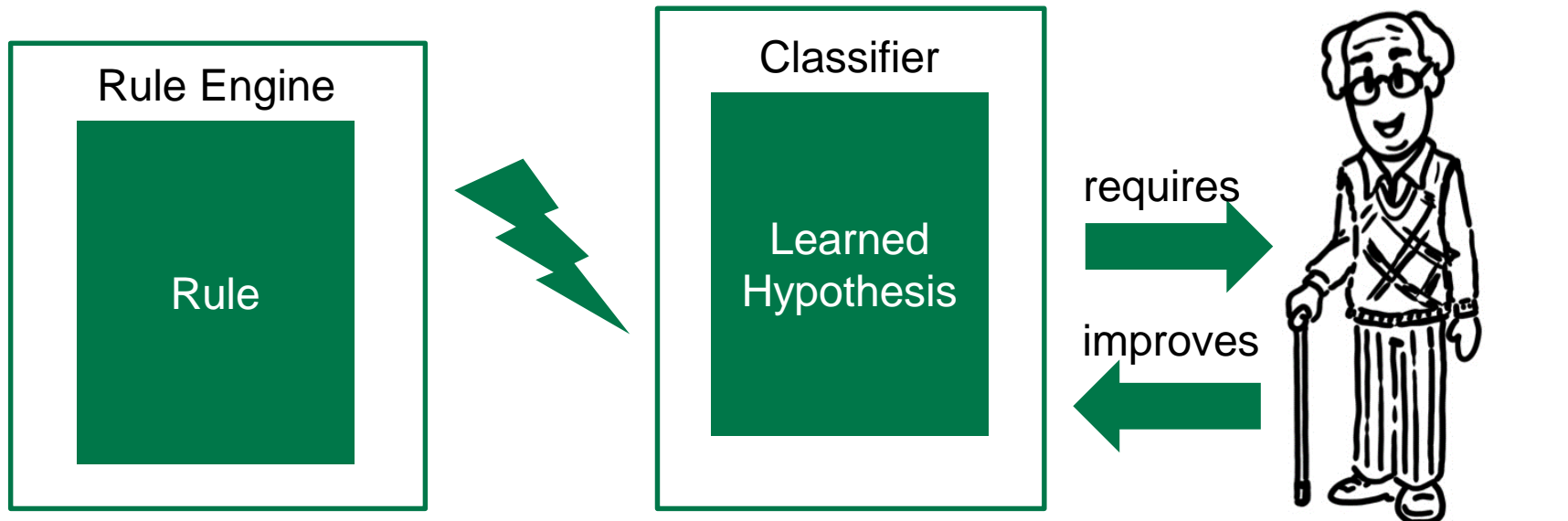
# Machine Learning Datasets based on SMW knowledge

hasState(?lamp, ?state)	hasState(?ambient, ?lux)	hasInFocus(?etg, ?thing)	triggeredAction(?user, ?action)
On	400	Lamp	SwitchOff
Off	600	Lamp	SwitchOn
Off	200	Fridge	-
On	100	Oven	-
On	700	Lamp	SwitchOff
Off	250	Lamp	SwitchOn



# Learning the Datasets by Naive Bayes Classifier

- Bayes Theorem:
  - $P(\text{action1}|\text{evidence}) = (P(\text{evidence, action1}) * P(\text{action1})) / P(\text{evidence})$
  - $P(\text{action2}|\text{evidence}) = (P(\text{evidence, action2}) * P(\text{action2})) / P(\text{evidence})$
  - Action1 if  $P(\text{action1}|\text{evidence}) > P(\text{action2}|\text{evidence})$  else Action2
- SWRL Rule:
  - **Evidence -> Action**



# Summary and Outlook

- **Embedded queries and rules** for programming
- Every Sherlock agent just needs to implement the **rule APIs**.
- The rules provide the **basis for learning the context** and user intention
- **Machine Learning** can be applied together with rule-based reasoning in order to **solve uncertainty**.
- The **Web of Things** approach provides further information to the agents about the IoT Things in the environment.

## Outlook:

- Implementing/Re-using a SMW Bot for adjusting the given rules.